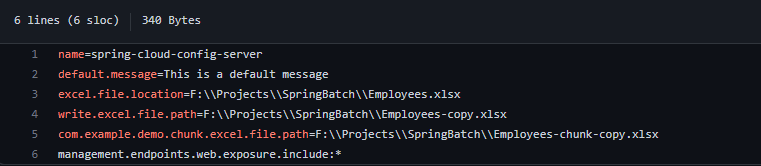
**TEST EVIDENCE- SPRING 12- Dynamic config update**

**Properties specified in config-repo:**

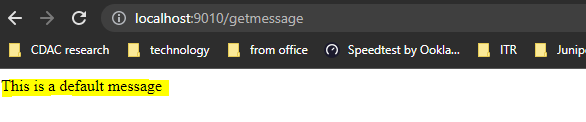


***Later on changed the field from default.message to message and moved the uri from application.properties to bootstrap.properties.***

**Endpoint created in client application:**

public class SampleController {  
@RestController  
@RefreshScope  
@Slf4j  
public class SampleController {  
  
 @Value("${message}")  
 String defaultMsg;  
  
 @GetMapping("/getmessage")  
 public String loadMessage() {  
 *log*.info("checking runtime config values: " + defaultMsg);  
 return this.defaultMsg;  
 }  
  
}

**when we call this endpoint we get the value in config file:**



**Now change the config to fetch changes on runtime without restarting the application:**

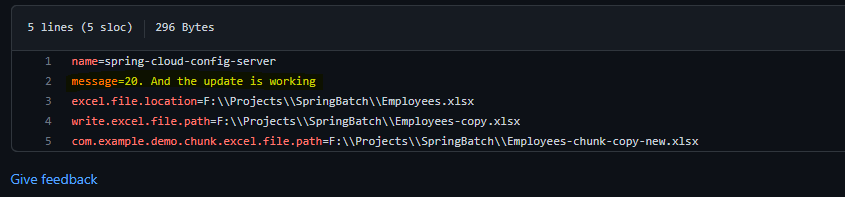
**Dependency added of actuator:**

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-actuator</artifactId>  
</dependency>

**Annotation refreshScope added to all the files which are fetching the properties from config-repo:**

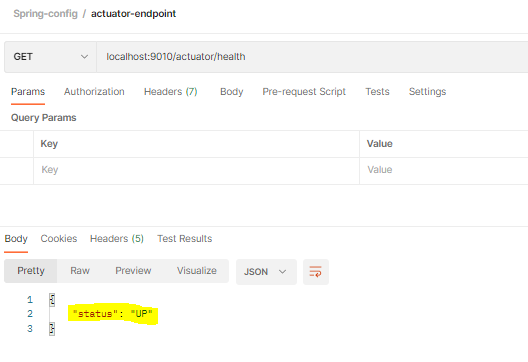
@RefreshScope

**Updated the message in properties file of config-repo:**

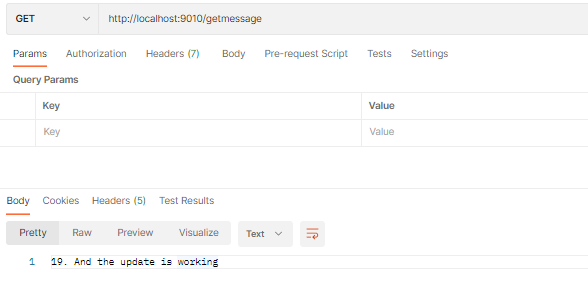


**We need to do a post call so that our client application’s properties gets updated.**

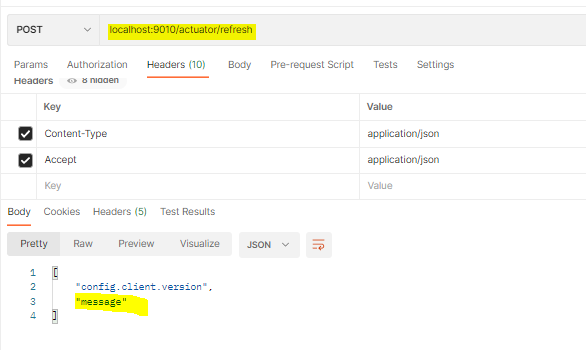
**But first see the health check**



**Before updating the config-client using /refresh POST call**



**Called the /actuator/refresh endpoint:**



**Called the /message rest controller endpoint:**

